基於AGM多標籤景觀聚類旅遊路線規劃

國立中正大學 National Chung Cheng University
學生：常博愛 蔡嘉祥　　指導教授：李新林

摘要：

由于目前的旅游景點的分群方法多基于單一標簽分群，會導致呈現出的景觀特質單一；同時通過逐個閱讀簡介並手動規劃旅游路線也爲游客帶來許多不便。因此本專題的主旨為，在多標籤景點下自動為遊客推薦一條含有多種景點的短期旅遊路線。我們利用觀光局提供的台灣旅游景點資料集，使用預訓練BERT模型將景點簡介向量化，再采用overlapping community detection的演算法將BERT模型微調成可以進行多標籤分類的任務，進而給各景點上多種標籤。同時依據OpenStreetMap(OSM) data來得到景點之間的行車時間。最後，在考慮景點標籤種類及行車時間這兩因素之下，給出一條優先考慮路綫內標籤種類盡量多的同時又兼顧平衡各種類出現次數與縮短行車時間的短期旅游路線。

方法與架構:

1. 多重標籤

1.1 模糊分群(Fuzzy clustering)

AGM (Graph Affiliation Model)

Objective function : negative log likelyhood

$$P(\mathrm{F}|\mathrm{G}) = -\sum_{(u,v)\in E}\log(1-e^{-F_uF_v}) + \sum_{(u,v)\notin E} F_uF_v$$

$F^* = argmin(P(\mathrm{F}|\mathrm{G}))$, $C_{n_j} = \{c_i \,|F^*_{n_j}[c_i] > thr., \forall c_i \in C\}$

F_u ： 為點 u 屬於各個社群的「強度」。

此公式是由機率模型推導而出，不過也可以直觀的理解：F_uF_v：u,v在經過 F的嵌入後兩點所屬社群的相似度。在圖上，若$(\mathrm{u},v) \in E, (u,v') \notin E$，則$u,v$屬於同一個社群的機率應該要比$u,v'$的機率來的高。而在最小化$P$的過程中，若 $(\mathrm{u},v) \in E$ ，則會傾向把F_uF_v變大；若 $(\mathrm{u},v) \notin E$則會傾向把F_uF_v變小，符合我們對community的期望。

- ***可以用來表示 overlapping community 的結構***

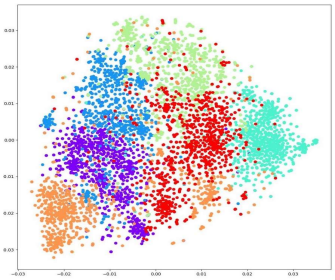
我們將景點簡介使用以語意相似度為目標的預訓 BERT 模型 SentenceTransformers 進行向量化，並依據該些向量的 cosine similarity 建立 KNN graph (k = 5)。我們使用兩層GCN (Graph Convolution Neural network)搭建AGM，將向量化之簡介與KNN graph 降維到分群數量之維度。藉由最小化 negative log likelyhood 來優化GCN內部參數。訓練好後的AGM即為分群的模型。

分群模型:

$\boldsymbol{ReLu(GCN(ReLu(GCN}(X,\hat{A}),\hat{A}))$

- $\hat{A} = \mathrm{D}^{-1/2}\mathrm{A}\mathrm{D}^{-1/2}$
- thr: 0.5

依照分成群數所能達到的 modularity之最大值，我們選擇分成6群



分群(6群)結果(t-SNE降維)
modularity : 0.63

1.2 群內分析

使用TF-IDF關鍵字擷取技術，針對每群內之景點的簡介提取關鍵詞，並依據重複多次的關鍵詞對該群下有意義的標籤。

Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster5
自然探索	山野農情	文化信仰	藝術人文	海湖風光	歷史紀念

2. 路徑規劃

依據所標之標籤，結合OSRM API所提供之行車時間計算，給出一條移動時間在一定範圍之內，所經標籤總類最多且出現次數平均的路線。以下路線評分公式，越大越符合我們所期望的推薦路線。

$$n + e^{-\frac{Var(\#Labels)+w_t * T}{10}}$$

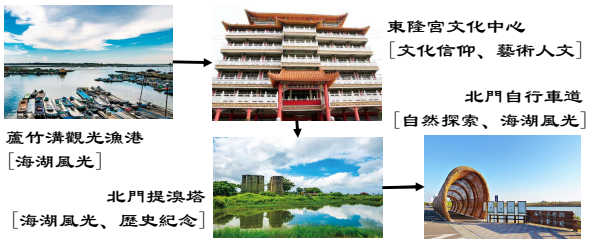
n:所經過之標籤種類個數 $\#Labels$：所經過之各種類標籤的計數向量
$Var(\cdot)$: $Variance$ T：總行車時長 w_t：正規化參數，我們選擇5

Algorithm Generate recommend path

cur := source
$Path := \{\emptyset\}$

while $|Path| < WantedNumber$ **do**
  Candidate ← driving time from cur to them ≤ time limitation
  n_{max} ← **max**(# nonzero entry of $Labels + c_i.Label$)
  Targets ← **argmax**$_{c_i}$ (# nonzero entry of $Labels + c_i.Label$),
    $\forall c_i \in$ Candidate
  if n_{max} dosen't imporve **then**
    target ← **argmin**$_{c_i}$(**Variance**($Labels + c_i.Label$)+d(cur, c_i)),
      $\forall c_i \in$ Candidate
    Update $Path$, $Labels$, total drving time, cur
    continue
  end if
  target ← **argmin**$_{t_i}$(**Variance**($Labels + t_i.Label$)+d(cur, t_i)),
    $\forall t_i \in$ Targets
  Update $Path$, $Labels$, total drving time, cur
end while

結果（路線範例）：



0	1	2	3	4	5
1	0	1	1	3	1

經過種類 : 5　Var : 0.805
行車總時長: 0.18 hrs (10mins)
Score: 5.176

0	1	2	3	4	5
1	2	1	1	1	1

經過種類 : 6　Variance : 0.138
行車總時長: 0.35 hrs (21mins)
Score: 6.154

結論

透過我們設計的這項推薦演算法，我們給使用者提供了豐富而多元的短路程旅遊參考路線。比較可惜的地方是，我們的分群模型並未到很完美。我們認為，如果可以針對資料集訓練出專屬的NLP模型，或許可以更進一步改善旅遊景點標籤的合理性，進而改善整體結果。